Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Smart extensions to regular cameras in the industrial environment

S. Brezani[1,*], R. Hrasko[1], P. Vojtas[1,2]

[1]*Globesy ltd, Framborska 58, Zilina, Slovak Republic*

[2]*Dpt. Software Engineering, Charles University, Malostranke nam. 25, Prague, Czech Republic*

Abstract

Data mining from unstructured data can be skillfully employed to improve the performance of manufacturing or industrial processes. The main goal of this paper is to create a system for object detection in industrial premises. We use several models of deep neural networks pre-trained for smart city applications. Our system works without retraining, additional annotation, or human intervention. Specifically, we present heuristics for the automated creation of PGT (Pseudo-Ground Truth) as a new form of unsupervised learning. Based on PGT, we can automatically decide which model is the best in the specific environment. We present an application of fully automated enhancing image capture camera outputs to smarter ones. We evaluate our system in a controlled experiment. We identify challenging videos and present a proof-of-concept for improving our system. The benefit is a knowledge extraction in a simple and inexpensive way to expand the organizations' databases with information from unstructured data from CCTV/IP cameras.

Keywords: Unsupervised learning; performance improvement of industrial processes; object recognition; pseudo-ground truth.

1. Introduction with contributions

Our primary interest in this paper is to use data mining on unstructured data to improve the performance of manufacturing or industrial processes. The main goal of this paper is to transform industrial premises images into helpful knowledge for decision-making without retraining them for this specific environment. We use a variety of deep neural networks pre-trained in a smart city environment. Specifically, we use object detection to recognize, e.g.,

* Corresponding author. Tel.: +421 911 802 025.
E-mail address: samuel.brezani@globesy.sk

humans from regular camera images. The use-case is a customer using low-quality cameras on its premises. Moreover, we assume the customer needs fast help in extending the functionality of his camera system without any annotation, training, or human intervention.

Furthermore, we consider that customers have no skilled human potential for manual annotation of data. We present an application of fully automated enhancing image capture camera outputs to smarter ones. The benefit is a knowledge extraction simply and inexpensively to expand the organizations' databases with information from unstructured data from CCTV/IP cameras. Another significant trend is to optimize the energy efficiency and speed of object detection models. This trend makes it possible to use these models directly on the data source - edge computing. With this concept, it is possible to augment each CCTV/IP camera with smart features and thus enable data mining of unstructured data for Industry 4.0 needs.

The main contributions of our paper are:

- Selection of variable data sources (video recording from different industrial environments)
- Selection of a variable system of models (pre-trained deep neural networks for object recognition task)
- Fully automatic system enhancing a standard camera system with object recognition based on
  - heuristic system for unsupervised creation of pseudo-ground truth
  - automatically choosing the best model for each environment
- Experiments evaluating the precision of our system
- Proof of concept of repairing weak results on "difficult" data

2. Videos with industrial premises and pre-trained models

Our goal is to use existing trained models of machine learning without any additional training to obtain "smart" information in the organization from their CCTV / IP cameras. We focus on ordinary cameras that do not support smart functionalities, resp. no "smart" software is supplied. The benefit of our solution for the organization could be a cheap and straightforward way to expand the organization's database with information from unstructured data from CCTV / IP cameras. The organization can use this data to control premises, parking lots, identify the use of premises, and trigger alarms, and the like. With more sophisticated analysis, an organization can identify trends in various events or use predictive analysis to predict future events. Some knowledge extraction should also be possible in the future.

2.1. YouTube videos with industrial environment

We focus on real examples of regular CCTV / IP cameras from industrial or business practice (for a few, see Fig. 1, more are depicted through the paper). For our research, we selected data from the youtube.com portal. We selected 13 videos from different environments (office, production line, parking, etc.) to obtain high variability. Description of abbreviated names and video properties is listed later; see Table 3. Full links to videos are in appendix Table 4.

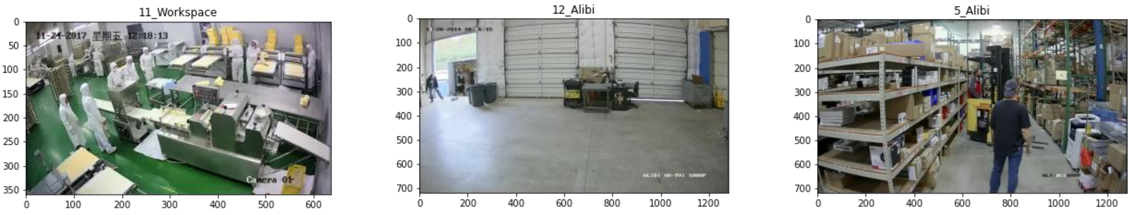

Fig. 1. Examples of regular images from CCTV / IP cameras.

2.2. Choice of deep neural networks pre-trained for object detection

A significant number of pre-trained computer learning models from the computer vision category are freely available on the Internet. Our primary interest is the area of industry. However, we did not find any training set from the industrial domain, e.g., office, production line, warehouse, parking lot, etc. Obtaining annotated data needed for training is a time-consuming activity that requires user intervention, and therefore, we want to avoid it. We chose several pre-trained models based on the COCO reference dataset [10]. This dataset contains more than 330,000 images containing annotations of 1.5 million instances of objects from 80 classes. But we can say that it is more from the area of "smart cities" and none of our target areas. Of course, we are aware that the accuracy of such models is probably significantly worse than in the case of their training in a specific environment (warehouse, office, etc.).

Many of these models use different architectures of deep neural networks. In our research, we selected models implementing the object detection task. This task identifies a set of instances of the objects in the image. For each object instance, the model returns:

- class - person, car, and the like
- bounding box - position and size of the object - boundary where the object is located
- confidence score with which the model predicts an object in a given bounding box.

Fig. 2 shows an example of prediction, where the image contains 2 classes of objects of our interest - 6 cars and 3 people (ground truth). Example of model identification: was able to identify 5 cars but/and no person (false negatives) while identifying one false positive object - a bird.

To achieve better representability, we chose 9 deep neural networks based on different architectures, backbone layers and process different sizes of input images for our research (see Table 1).

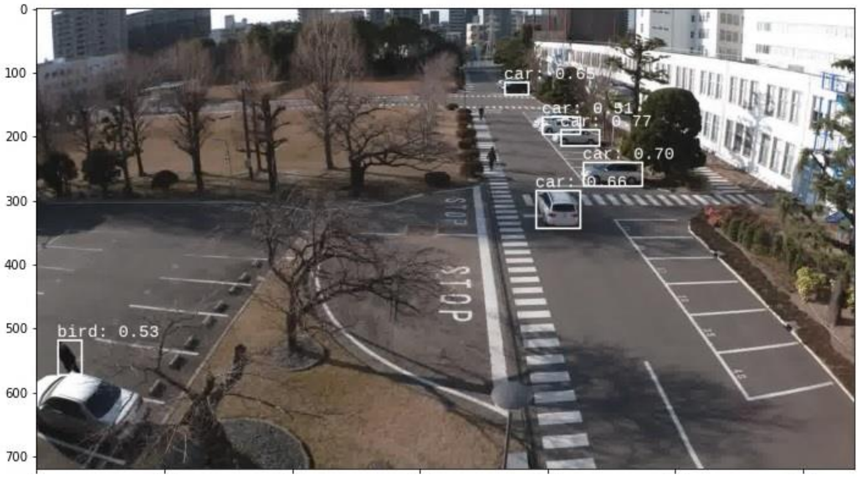

Fig. 2. Illustration of object detection by a neural network.

Table 1. Chosen models, backbone, and the number of identified object instances and classes for the model/confidence level.

	Confidence	cf > 0.3		cf > 0.5		cf > 0.7		cf > 0.9	
Backbone	Model / predicted	inst	cat	inst	cat	inst	cat	inst	cat
Darknet53	YOLO4	13760	39	10366	27	7942	16	4486	10
CSPDarknet53	Retinanet-ResNet50	23910	67	9800	36	4646	18	309	6
ResNet50	CenterNet-HourGlass104	21939	60	9714	40	3476	16	185	3
ResNet101	CenterNet-ResNet50	14419	53	4527	20	849	8	9	1
ResNet152	EfficientDet D3	20954	52	9883	27	4321	14	252	2
HourGlass104	EfficientDet D6	27556	47	11507	31	5807	14	344	3
ResNet50	Faster R-CNN	19947	70	12026	58	8343	46	4299	18
EfficientNet B3	Retinanet-ResNet101	23712	62	6953	35	1275	7	7	3
EfficientNet B6	YOLO3	13935	42	9344	34	7118	21	4802	11

3. Fully automated creation of pseudo-ground truth

3.1. *Model analysis*

For our needs, we extracted images from the video (frames with a frequency of 1 per sec). This parameter is discussed later – it can influence the quality of our service. Using all models, we performed predictions on each of the 3838 images (see Table 3 on video length). Table 1 shows model ID with backbone architecture and the number of identified instances and classes of objects for each model at different confidence levels (sum on all videos). Note that we are not talking about correctness - we take the position of correctly marking classes and instances based on the observed legalities – to have a fully automated solution.

Nevertheless, we can see that the simple use of neural networks and simple voting is not working. Networks trained on smart city data recognize too many classes and instances that clearly do not fit into the industrial environment (as proof of concept, we have experimented with semantic methods to recognize classes that do not fit into the premise).

3.2. Our solution – automated creation of pseudo ground truth

An important criterion for our method is to minimize the need for human intervention and support the automation of such a process. Based on the established criteria, we proposed a method for identifying the most suitable machine learning model from N available models. The frame is classified by all models and we design a "merge prediction" method deciding which instances with classes will be chosen to our ground truth. We extract images from the input video at the selected frequency (in our case, 1 second). For each image, we make a prediction using each model. This way, we get a set of identified objects from all models from the input image. For further processing, we select only those predictions whose confidence > 0.5.

Because different models assign different bounding boxes to an instance, our heuristic assumption is (if correct) that these bounding boxes are not very different. The next step is to combine these predictions using our algorithm to obtain a "pseudo ground truth". We call it "pseudo" because it is created fully algorithmically, without any human intervention during annotation. The algorithm for combining predictions to form "pseudo ground truth" is something like this:

```
pairs_queue = []
bbox_group = {}
FOR bbox_a IN bboxes:
        FOR bbox_b IN bboxes:
                IF bbox_a != bbox_b && iou(bbox_a, bbox_b) > treshold: pairs_queue += (bbox_a, bbox_b,  IOU(bbox_a, bbox_b))
SORT(pairs_queue, KEY=2, DESCENDING) # descending sort by iou values
FOR idx, bbox IN ENUMERATE(bboxes): bbox_group[bbox_a] = idx
FOR [bbox_a, bbox_b, iou_score] IN pairs_queue: ga, gb = bbox_groups[bbox_a], bbox_groups[bbox_b]
        IF ga == gb: CONTINUE
        merge_candidates = SET([bbox IN bboxes IF bbox_groups[bbox] IN (ga, gb)])
        model_candidates = SET([MODEL(bbox) FOR bbox IN merge_candidates])
        IF COUNT(merge_candidates) != COUNT(model_candidates): CONTINUE
        FOR bbox in merge_candidates: bbox_group[bbox] = ga
```

Algorithm inputs are all frames from a video. For each frame separately, the algorithm searches for all prediction pairs whose IOU - intersection over union metric is greater than the threshold parameter (in our case, 0.5). It stores such pairs in the pairs queue field, sorts these pairs in descending order according to their IOU value. Then the algorithm initializes a key variable bbox_group representing each prediction assignment. Subsequently, the algorithm iterates through the pair of predictions. Let us comment here that this algorithm is a pure heuristic. We do not aim to have a correct algorithm because we do not have annotated ground truth. We aim to create a pseudo-ground truth formed by model candidates automatically.

In the next step, we initialize the set of all predictions that belong to the group. These are candidates for merge – elements of future pseudo ground truth. There are 1 to N predictions in each group (N is the number of models), each coming from a different model. Because only those predictions whose IOU > threshold (which we consider to be a sufficient condition for "identity" of objects) are combined into groups, we consider all predictions in a group to be the identification of the same object.

Fig. 3 shows an example of combining predictions that we achieved with the algorithm. The image on the left shows color-coded predictions from different models represent how many of the models identified each object. The right part of the figure shows the already associated predictions together with the score (number of models agreeing). Fig. 4 shows a more complicated situation in which two people overlap, our algorithm even in this case combined the predictions correctly.

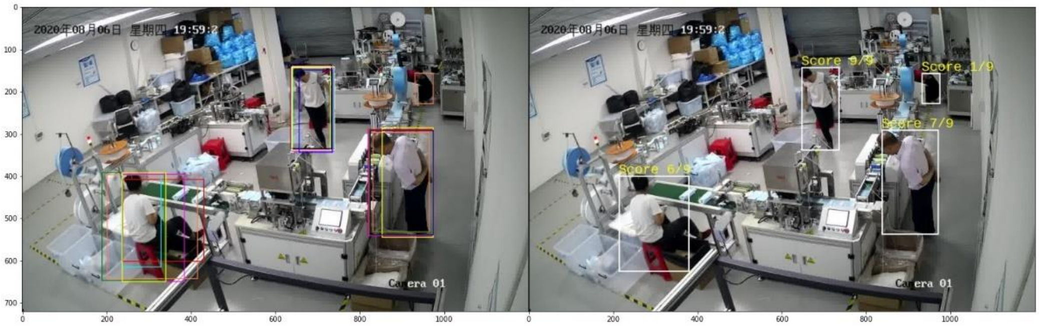

Fig. 3. Left bounding boxes (bbx) of various models, right bbx with number of models which object recognition agrees – these form an algorithmically created "pseudo ground truth" (without any human intervention)

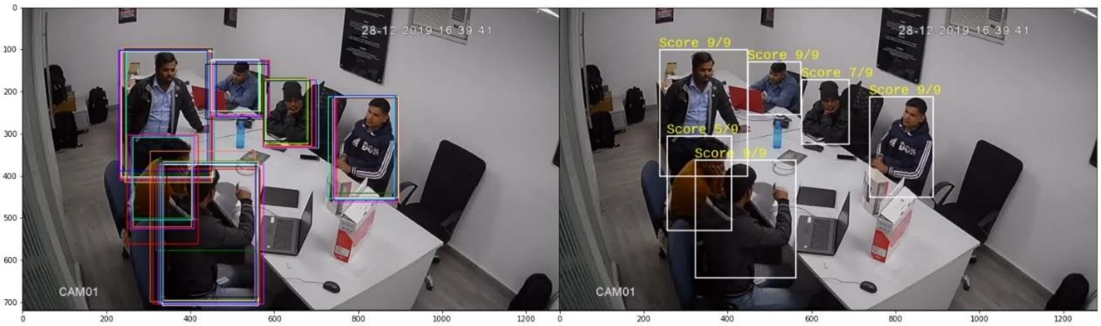

Fig. 4. More complicated situation in which two people overlap. Even in this case, the algorithm combined the predictions correctly.

After processing the image according to the algorithm, we obtain grouped predictions for each image. We call them pseudo-ground truth. For each such prediction, we know how many of all the models identified a given object. At this point, we define a key assumption: if there is a real object in the image, then we assume that at least K of the N models identifies this object. In our research, we worked with N = 9 models and chose K = {3, 5, 7}. It should be noted that in this heuristic, we do not consider at all the confidence with which the individual models recognize the instances - and the groups could also be assigned "group" confidence. Using this assumption, we obtain a set of pseudo ground truth (PGT_K) objects for each image and K = {3, 5, 7}.

3.3. Choosing the best model by the highest recall

We already have all the necessary input data - images and identified objects to select the most suitable model. For each K = {3, 5, 7} and each model, we identify the objects in the images and compare them to PGT_K. The main point of our selection of the best model m_K is that we chose the one with the highest recall (over all frames, measure see [12]). We find this an important part of our heuristics because we would like our customers to have rather higher recall than higher precision. This also lowers the false negatives rate.

To our surprise, the best results were obtained for K = 3 (in the sense that recall was higher than for K = 5, 7). Nevertheless, we wanted to evaluate our method more precisely; see next chapter.

4. Validation

To validate the accuracy/precision of our heuristics, we manually annotated a portion of the input data for each video. We then performed object identifications for the annotated data by all models with a confidence of at least 0.5. Subsequently, we compiled a list of models for each video, sorted by average precision ([12]) over all frames. For comparison, we also evaluated these metrics for the models recommended by our heuristic and human-annotated GT.

Table 2. Comparison of automated PGT versus human-annotated GT and best model metrics.

	Computed PGT_3 / human-annotated GT		The best model m_3 performance on human-annotated GT		
Video source	Average Precision	recall	Average Precision	recall	Best chosen by our system ID
0_CCTV	0.9944	0.9944	1.0000	1.0000	YOLO3
10_MAGGI	0.8235	0.8235	0.9108	0.9118	EfficientDet D6
11_Workspace	**0.1856**	**0.1926**	**0.2664**	**0.2704**	**EfficientDet D6**
12_Alibi	0.6984	0.6984	0.6974	0.6984	EfficientDet D6
13_SLEEPING	**0.2000**	**0.2000**	**0.2000**	**0.2000**	**YOLO4**
2_4K	**0.3329**	**0.3425**	**0.3650**	**0.3836**	**EfficientDet D6**
3_BMW	1.0000	1.0000	1.0000	1.0000	Faster R-CNN
4_APL80	0.8199	0.8288	0.6089	0.7055	EfficientDet D6
5_Alibi	0.9592	0.9592	0.9592	0.9592	Faster R-CNN
6_HD	1.0000	1.0000	0.9133	0.9286	Retinanet-ResNet50
7_Allied	0.6452	0.6452	0.5914	0.5914	EfficientDet D6
8_Warehouse	0.5000	0.5000	0.5000	0.5000	EfficientDet D6
9_MESSOA	0.8115	0.8333	0.7976	0.8333	CenterNet-HourGlass104

The bold rows in Table 2 show data sources with a low average precision value (<0.4). If we look at the videos' quality attributes, we can get some hints where our method is weak depending on video properties, see Table 3.

Table 3. List of video properties used for evaluation later. See Appendix for more video information.

Short name	Premise	Frames	Resolution	Area size	Camera type	Number of persons
0_CCTV	office	31	high	small	static	few
10_MAGGI	manufacturing	174	high	small	PTZ	few
11_Workspace	manufacturing	182	low	medium	static	many
12_Alibi	warehouse	55	high	medium	static	few
13_SLEEPING	warehouse	34	low	small	static	few
2_4K	parking	275	high	large	static	many
3_BMW	manufacturing	858	high	small	PTZ	few
4_APL80	manufacturing	1409	low	medium	static	many
5_Alibi	warehouse	29	high	small	static	few
6_HD	office	193	high	small	static	few
7_Allied	warehouse	178	high	large	static	few
8_Warehouse	warehouse	265	high	small	static	few
9_MESSOA	warehouse	155	high	small	PTZ	few
	Total	3838				

The area size and resolution defined in Table 3, we can visualize these data sources using the graph in Fig. 5. We can see the dependency between resolution, scene area size, and average precision value from the graph. A more detailed analysis is a goal in our further research. Some initial experiments are described in the Conclusions.

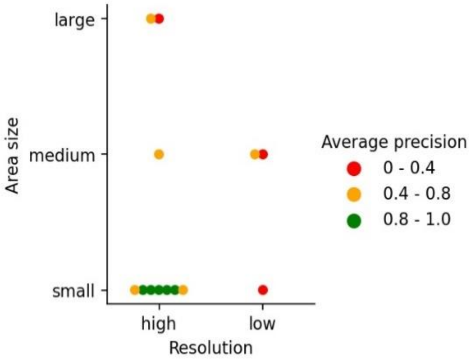

Fig. 5. The dependency between video attributes and the average precision value of the models selected by our heuristic.

5. Related work

In the context of our work, it is also important to mention the evolution and variability of object detection models because our method benefits precisely from their variability. Subsequently, we discuss similar solutions.

Object detection methods have been the subject of research for more than 20 years, see, e.g. [19], and are now long established in industrial practice. They are used in many fields such as face detection, pedestrian detection, traffic sign detection, text detection, etc. Research in this area has focused on several directions.

The first is improving object detection models themselves. One of the basic architectures is R-CNN architecture. [7] was with the first major deep neural network methods for the task of object detection. Other architectures such as [8] and [17] are an improvement on the original R-CNN architecture, succeeding in significantly increasing the speed of the model. Another breakthrough architecture was [16], which was the first to use the concept of one-step object detection. This concept brought significantly higher model speed but at the expense of accuracy. Other architectures such as [1] and [18] improved on the original YOLO architecture while optimizing speed and energy efficiency. Optimizing object detection models' energy efficiency and speed allows them to be used directly at the data source (edge computing). Using this concept, it is possible to extend each CCTV/IP camera with smart features and thus enable data mining of unstructured data for Industry 4.0 needs. Nevertheless, they all need an annotated data set to be trained.

Another research direction focuses on specific improvements to selected problems of existing methods. For example, the method [2] focuses on minimizing false positive detections of nearby or overlapping objects.

Other methods focus on eliminating hard false positives, see, e.g. [4] and [3]. These methods are based on extending the detection model with an additional classification model.

We also identified problems with hard false positive and false positive detections of nearby and overlapping objects during our research. However, in our case, we could not use these methods because they are from the supervised learning category. Our goal is to find a method that does not require training or human intervention to select a machine learning model for a particular data source. The problem of selecting a model or the values of its hyperparameters is part of the iterative process of training the model. Related to the model selection is the phase of its evaluation using the selected function (accuracy, recall, f score, etc.). In the case of the object detection task, the evaluation function average precision is often used see, e.g. [13].

Model selection and its evaluation are most often accomplished by training and evaluating the model on disjunctive datasets - the training and validation sets. These sets are generated using methods such as K-Fold, LOO, LPO, and others ([14]). In our case, we cannot use these methods directly because we need the ground truth to evaluate the model using them.

Further research in the field of AutoML has also brought the possibility of automatic model selection. Framework Auto-Keras [9] uses Bayesian optimization to select the most appropriate neural network architecture.
Framework Auto-Sklearn [5] uses various meta-learning techniques not only to identify the best model but also to build a pipeline for data preprocessing. Similar to the previous case, we cannot use these methods directly because they are from the supervised learning category as they require ground truth data. As can be seen, for selecting the best model, a set of ground truth data is required. Our method introduces a heuristic by automatically generating a set of "pseudo ground truth" data.
Our method uses a similar idea to learning from crowds methods as [15], just our crows is a crowd of deep neural networks. These methods are based on the principle that multiple annotators - experts with different experiences annotate the same data. In this way, we obtain multiple subjective sets of ground truth data. Subsequently, these methods approximate ground truth by different approaches - majority voting, maximum likelihood estimation etc.

In [11] authors show how to utilize data mining techniques which are now within reach of numerous smaller manufacturing operations. Moreover, they provide a further understanding of how moves towards fully Industry 4.0 ready factories may be made in the years to come.

In [6], authors review existing data mining (both unsupervised and supervised) and analytics applications in the process industry over the past several decades. They highlighted and discussed several perspectives for future researches on data mining and analytics in the process industry.

6. Conclusions, future work, and plans for deployment

Our research has provided a method for selecting an object detection model for a particular data source. In our method, we used a simple mechanism - based on the principle of learning from crowds to generate a set of pseudo ground truth data. We expect that by using more sophisticated methods for generating pseudo ground truth, the accuracy of selecting the best model can be improved. We hope the benefit is a knowledge extraction in a simple and inexpensive way to expand the organizations' databases with information from unstructured data from CCTV/IP cameras. Thanks to the new paradigm of Industry 4.0 and by employing information technology, this process becomes easier.

A challenge remains false negative. Optimizing our method on recall lessens this problem. An interactive deployment can help. E.g., giving a permanent operator the possibility to comment on our output (just clicking what is correct-true positive, what not-false positive, what is missing-false negative etc.).

We have to have prepared default models for initial functioning from time t = 0 on, for various industrial environments. Simultaneously, we start to record a video stream. We select a time for learning t_1 where a sufficient period with typical operation repeatedly occurred. Let us run the above algorithm on these records and select K with the best result on PGT_K. We select the best model, m_K, and start to use it for this video.

We intend to extend our method by continuously evaluating model selection on data from a customer. Consider a situation in which our method selects the best model for a particular data source. However, we will not stop with this step, but we will rescan for the best model at regular intervals (days, weeks). Using the data thus obtained, we can use unsupervised learning methods to change the model according to the typed scene (day, night ...).

This extension of our method would bring the possibility of automatically changing the object detection model for a specific CCTV/IP camera. We envision that this extension would be mainly applicable when the external environmental conditions change periodically, such as the alternation of natural light and artificial light/night mode of the camera. Additionally, this extension will also bring the ability to adapt to sporadic changes - moving the camera, significant scene change, etc.

We consider improving our object detection method in case of poor results on some videos (e.g., three videos in Table 1). This can also be detected automatically in a deployed situation. The idea is not to process the input data in isolation but as a sequence of frames (perhaps with a higher frame rate, e.g., 0.2 seconds). Thus, we obtain spatio-temporal data. In this case, we have object identification not only for the current situation but also for the short past and the future (a window of 5 frames would be sufficient). A buffer with object identification from the window frames would serve for additional correction of the current prediction. Some initial experiments were provided, so as a proof of concept, it is promising. We do not present it here. Large areas or inadequate lighting conditions cause some mistakes, so some distant objects are misclassified. Improving lighting conditions, installing more cameras, or requiring object detection only for the closest camera are possibilities for avoiding it. We are working on it and gaining experience.

Semantic analysis of identified objects with respect to the environment could also be an interesting approach for removing false positive predictions. In the case of processing big data from a large number of typed sources (parking lot, office, warehouse, …) we could identify the probability of an object's occurrence in a given environment. This approach would be suitable for identifying semantically unacceptable objects in a particular environment - a car in an office, an elephant in a parking lot, etc. Specific knowledge extraction to augment the organizations' databases with information from unstructured CCTV/IP camera data using semantic methods is also left for future work.

Acknowledgments

This publication was realized with the support of the Slovak Operational Programme Integrated Infrastructure in the frame of the project: Intelligent systems for UAV real-time operation and data processing, code ITMS2014+: 313011V422 and co-financed by the European Regional Development Fund.

Appendix

Table 4. Full YouTube information of our data.

Short name	Link	Title on YouTube
0_CCTV	https://www.youtube.com/watch?v=mMXDU8q4fIQ	CCTV Office Surveillance Camera Office CCTV Surveillance
10_MAGGI	https://www.youtube.com/watch?v=cIVDZiJ7NjY	MAGGI Lean Assembly Line
11_Workspace	https://www.youtube.com/watch?v=xBZzc8iZjT8	Workspace accident - When two workers playing with the machine at the same time
12_Alibi	https://www.youtube.com/watch?v=O2pG39jO0JE	Alibi 1080p HD-TVI WDR Security Camera - Warehouse
13_SLEEPING	https://www.youtube.com/watch?v=ZdZeja401xM	SLEEPING IN FACTORY,24X7CCTV MONITORING,CCTV MONITORING,CCTV SURVEILLANCE,CCTV SECURITY,CCTV CAMERA
2_4K	https://www.youtube.com/watch?v=wqlO5i3N-FU	4K Camera in use at Panasonic Office in Japan
3_BMW	https://www.youtube.com/watch?v=P7fi4hP_y80	BMW Car Factory ROBOTS - Fast Manufacturing
4_APL80	https://www.youtube.com/watch?v=yZcufAb7sMk	APL80 Automatic Mask Machine - IP Camera View Part 1 - 20200806195858195 1
5_Alibi	https://www.youtube.com/watch?v=0xHvoobqHhE	Alibi 1080p HD-TVI Bullet Camera (ALI-BC1080P)
6_HD	https://www.youtube.com/watch?v=O6IMyzDEmHE	HD Camera Office Building WDR Performance
7_Allied	https://www.youtube.com/watch?v=CxpRrRBxl4Q	Allied Security Links Warehouse Megapixel Camera Image
8_Warehouse	https://www.youtube.com/watch?v=YQtBjpReiaM	Warehouse CCTV
9_MESSOA	https://www.youtube.com/watch?v=OO7XT24AmTY	MESSOA PTZ900-3MP Indoor PTZ Network Camera / Warehouse Surveillance

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection, *arXiv.org > cs > arXiv:2004.10934*

[2] Z. Cai and N. Vasconcelos. (2018) Cascade R-CNN: Delving Into High Quality Object Detection, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6154-6162, doi: 10.1109/CVPR.2018.00644

[3] B Cheng, Y Wei, H Shi, R Feris, J Xiong, T Huang. (2020) Decoupled Classification Refinement: Hard False Positive Suppression for Object Detection. *arXiv preprint arXiv:1810.04002*

[4] Cheng B., Wei Y., Shi H., Feris R., Xiong J., Huang T. (2018) Revisiting RCNN: On Awakening the Classification Power of Faster RCNN. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) In *Computer Vision*, Lecture Notes in Computer Science, vol 11219. Springer, Cham. https://doi.org/10.1007/978-3-030-01267-0_28

[5] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, F. Hutter. (2020) Auto-Sklearn 2.0: The Next Generation. *arXiv.org > cs > arXiv:2007.04074*

[6] Z Ge, Z Song, SX Ding, B Huang. (2017) Data mining and analytics in the process industry: The role of machine learning - *Ieee Access*, - ieeexplore.ieee.org

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, doi: 10.1109/CVPR.2014.81.

[8] R. Girshick. (2015) Fast R-CNN, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[9] Haifeng Jin, Qingquan Song, Xia Hu. (2019) Auto-Keras: An Efficient Neural Architecture Search System. *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Pages 1946-1956 https://doi.org/10.1145/3292500.3330648

[10] T-Y. Lin et al. (2015) Microsoft COCO: Common Objects in Context. *arXiv.org > cs > arXiv:1405.0312*

[11] Oliff H. , Liu Y. (2017) Towards Industry 4.0 Utilizing Data-Mining Techniques: A Case Study on Quality Improvement, *Procedia CIRP* **63**: 167-172.

[12] R. Padilla et al. (2020) A Survey on Performance Metrics for Object-Detection Algorithms," *Int. Conf. Systems, Signals and Image Processing (IWSSIP'2020)*, pp. 237-242, doi: 10.1109/IWSSIP48289.2020.9145130, for more see Object-Detection-Metrics on Github

[13] Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. (2021) A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **10**(**3**) 279. https://doi.org/10.3390/electronics10030279

[14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P. , Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011), Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, **12**, 2825—2830, see also further versions of Scikit-learn on Github

[15] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, Linda Moy. (2010) Learning From Crowds, *Journal of Machine Learning Research* **11**(**43**):1297−1322,

[16] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. (2016)You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[17] S. Ren, K. He, R. Girshick and J. Sun. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(**6**): 1137-1149. doi: 10.1109/TPAMI.2016.2577031.

[18] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao. (2021) Scaled-YOLOv4: Scaling Cross Stage Partial Network, *arXiv.org > cs > arXiv:2011.08036*

[19] Z Zou, Z Shi, Y Guo, J Ye. (2019) Object detection in 20 years: A survey, *arXiv preprint arXiv:1905.05055, 2019 - arxiv.org*